CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

PlanYourWay: A Smart Travel Itinerary Planner Based on Recommendation and Distance Optimization

By

Ashika James (2248032)

Ria Liz Luke (2248051)

Under the guidance of

Dr. Rupali Wagh

A Project report submitted in partial fulfillment of the requirements of III Semester Master of Science (Data Science) of CHRIST (Deemed to be University)

November – 2023

CERTIFICATE

This is to certify that the report titled **PlanYourWay: A Smart Travel Itinerary Planner Based on Recommendation and Distance Optimization** *is a bona fide record of work done by* ***Ashika James (2248032) and Ria Liz Luke (2248051)*** *of CHRIST (Deemed to be University), Bengaluru, in partial fulfillment of the requirements of III Semester MSc.(Data Science) during the year 2023.*

Head of the Department **Project Guide**

Valued-by:

	Name	: Ashika James
1.	Register Number	: 2248032
	Examination Centre	: CHRIST (Deemed to be University)
2.	Date of Exam	:

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

CERTIFICATE

This is to certify that the report titled **PlanYourWay: A Smart Travel Itinerary Planner Based on Recommendation and Distance Optimization** *is a bona fide record of work done by* ***Ashika James (2248032) and Ria Liz Luke (2248051)*** *of CHRIST (Deemed to be University), Bengaluru, in partial fulfillment of the requirements of III Semester MSc(Data Science) during the year 2023.*

Head of the Department **Project Guide**

Valued-by:

	Name	: Ria Liz Luke
1.	Register Number	: 2248051
	Examination Centre	: CHRIST (Deemed to be University)
2.	Date of Exam	:

ACKNOWLEDGMENT

It gives us a great sense of pleasure to present the report of the Project Work undertaken as part of our MSc Data Science course.

We owe special debt of gratitude to Dr Deepthi Das, HOD, Department of Data Science and Statistics, CHRIST (Deemed to be University) and our Project Coordinator Dr Dalwin Santhosh Kumar for their constant support and guidance throughout the course of our work. It is only their cognizant efforts that our endeavours have seen light of the day.

Our deepest thanks to our Project Guide Dr Rupali Wagh, Coordinator, Department of Data Science and Statistics, CHRIST (Deemed to be University) whose expertise, guidance, and mentorship have been the cornerstone of this project's success. Your patience, encouragement, and dedication to our development as a student and researcher have been truly inspiring.

We acknowledge the contribution of Greynium Information Technologies Pvt. Ltd, Bangalore for providing the data which is the soul of this project and led to the development of the project.

We are also grateful for the contribution of all faculty members of the department for their kind assistance and cooperation during the development of the project.

Lastly, we acknowledge our friends for their contribution in the completion of the project.

ABSTRACT

A travel itinerary is a plan of a journey showing the route and the places that the visitors will visit. It is basically a schedule or timetable produced based on number of days of travel and other necessary information. It is designed to identify the route, day-by-day journey format, origin, destination and all the enrooted halting points.

This academic project aims at developing an Itinerary planning web application that provides customers with itineraries based on Recommendation and Distance Optimization. The use of machine learning has given us a vast stretch to the solution to unsolved problems. One of the best ways to rank the top attractions in a place is by building a recommendation system. Here we use that technology along with distance optimization to build our application. We aim to build a basic recommendation system that identifies the topmost liked attractions of a place by using sentiment analysis.

The web application is built using Django Framework and makes use of Python in the backend. Further, MySQL has been used for handling the backend. There are three modules on the website, each of which are elaborated later in the report.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

The project aims at building a travel planning web application called PlanYourWay that generates an itinerary based on the place of interest and the number of days specified by the user. The application makes use of recommendations to filter the top attractions and optimizes the travel route through distance optimization. This chapter gives us a brief overview of the project along with the objectives, purpos,e and scope of this project.

1.1 PROJECT DESCRIPTION

South India is a vibrant and culturally rich land, where ancient temples, serene beaches, misty hill stations, and bustling cities has always been a centre of attraction for tourism in India. However, when it comes to tour planning, multiple factors like lack of proper information and language barriers make it challenging to plan the trip effectively. A travel itinerary planner exclusively for South India filtering the attractions based on reviews and ratings provided by other tourists and then optimizing the travel route will be highly beneficial to the user.

This project aims to provide a web application which consists of the following modules:

- Registration and Login (for customer and retailer/business stakeholder)
- Customer specification for the tour (place of interest, days of visit)
- Travel itinerary
- Explore destinations

1.2 EXISTING SYSTEMS AND THEIR LIMITATIONS

Travel itineraries are provided by many tour packages, but most of these are prepared manually. There are quite a few research papers with similar objectives. However, the approach used is different. A study conducted by Ivan Contoro and Sandy Kosasi [1] tackles planning routes from multiple locations, using genetic algorithms and the Google Maps API to improve efficiency by overcoming starting point limitations and reducing user uncertainty. This paper is entirely based on distance optimization. The major limitation identified in this paper is that the user is expected to know all the attractions of the place

he intends to visit. The application will then optimize the route based on the input locations provided.

Cost and time constraints were incorporated into the tourism route selection model based on an improved genetic algorithm (IGA) in the paper [2]. Here, the ant colony optimization (ACA) is used to overcome the limitations in similar works. The same problem mentioned in the first paper is seen here as well.

The paper [3] proposes an improved k-means cluster analysis algorithm that shows better clustering results compared to the traditional algorithm. The improved algorithm is applied to the design of an intelligent tourism route planning scheme that takes into account the tourists preference metric. The real route is not taken into consideration here. The clustering is done based on Euclidian distance between the latitudes and longitudes of the attractions. This is only very less accurate approximation and not the actual distance between the locations.

Another study conducted [4] proposes an artificial intelligence-based wireless sensor travel route planning study, which aims to design reasonable travel routes with the least time cost and the highest experience index. The paper addresses three main problems: classification of influence factors for the travel experience index, allocation of weights using a particle group algorithm-based projection tracing method, and the establishment of an optimization model for the highest tourism experience index. The paper presents the ND-GA algorithm, an improved version of the traditional genetic algorithm, to solve the mathematical model of energy consumption for wireless transmission in travel route planning. The algorithm is used to compute the improved TSP model and obtain 24 travel routes, with the satisfaction time being the highest, allowing visitors to visit all 2015A scenic spots.

Paper [6] is on building a travel planning system based on recommendations made using sentiment analysis, user preferences and previous reviews. The recommendation system is build using deep learning framework. However, the final output is not based on optimal travel route.

The uniqueness in this project is combining the power of recommendation and distance optimization that can provide the users with the most visited and liked attractions in a place in such a way that the whole trip can save time and cost since the distance is optimized.

1.3 OBJECTIVES

The objectives of this project are as follows:

● To generate a travel itinerary based on user specifications using recommendation and distance optimization techniques.

● To provide the model as a service to people by forming a web application for an easier and efficient travel experience.

1.4 PURPOSE, SCOPE AND APPLICABILITY

1.4.1 PURPOSE

In an era characterized by an abundance of choices, the need for personalized and efficient decision-making tools has become increasingly evident. This project seeks to address this need by developing a web application that leverages recommendation systems and distance optimization techniques. The primary objective is to assist travellers in crafting travel itineraries. With a wealth of destinations, points of interest, and limited time at their disposal, travellers often find themselves overwhelmed. This web application aims to simplify the travel planning process by offering tailored recommendations based on user preferences and optimizing routes for efficient exploration. By doing so, it intends to enhance the travel experience, saving time and energy while ensuring travellers can make the most of their journeys.

In the context of South India's diverse and culturally rich destinations, the project's focus is on creating a seamless and user-friendly platform. The application will not only recommend places of interest but also take into account the geographical proximity of these destinations, effectively reducing travel time and optimizing the overall itinerary. As a result, travellers can expect a hassle-free and memorable experience, allowing them to delve into the region's vibrant culture and natural wonders without the challenges of planning. In essence, this project aims to provide a solution that empowers travellers to make well-informed choices while exploring South India, transforming the travel planning process into a more enjoyable and efficient experience.

1.4.2 SCOPE

India is considered one of the most preferred tourist destinations in the world and attracts millions of tourists every year. The Ministry of Tourism reported that in 2022, over 6.19 million foreign tourists chose to visit India. Among the top five destinations are three districts of South India - Tamil Nadu, Karnataka, and Andhra Pradesh. South India is known for its traditional and cultural values, glistening beaches, and scenic landscapes that captivate travellers. However, when it comes to tour planning, multiple factors like lack of proper information and language barriers make it challenging to plan the trip effectively. Development of a Smart Travel Itinerary Planner can accelerate change by making travel planning more efficient, personalized, sustainable, and user-friendly.

Utilizing technology for travel planning, as demonstrated by this project, has the potential to revitalize the tourism industry. By offering personalized itineraries and efficient route optimization, it attracts more tourists seeking a hassle-free experience. This, in turn, can stimulate local economies by increasing footfall at tourist destinations, supporting local businesses, and creating job opportunities. The application empowers the tourism sector to cater effectively to traveller's needs, ultimately enhancing the overall tourism experience and contributing to the industry's growth.

While many travel planning tools do not cover many of the destinations in India, our project is exclusively designed for South India. This specialized focus allows us to provide in-depth and tailored recommendations specific to the rich and diverse attractions in India, catering to both domestic and international travellers.

1.4.3 APPLICABILITY

This project will be an easy to navigate and use web application that uses minimum hardware and software components for its use. This application would be extremely useful for the customers who want to save time and cost and visit the best attractions in a destination. The user-friendly UI of this application helps the users with even minimum technical knowledge to make use of all the functionalities offered and to navigate through the application easily.

1.5 NEED AND MOTIVATION

South India is known for its serene beauty, beaches, hill stations, spice plantations and many more. The major tourist destinations in South India include Goa, Pondicherry, Coorg,

Idukki, Wayanad, Bangalore, Tirupati, Hyderabad, Dhanush Kodi and many more. However, South India's geographical and cultural diversity presents unique challenges for travellers when planning their journeys. The sheer range of destinations and cultural differences can often lead to overwhelmed travellers who struggle to optimize their travel routes and make the most of their trips. This highlights the pressing need for an intelligent and user-friendly platform that assists in crafting personalized itineraries.

While Google offers an array of travel itineraries, the sheer abundance of choices often leads to confusion among travellers. Additionally, as social beings, people tend to rely on reviews to inform their decisions. However, sifting through countless reviews to determine which attractions to visit is a daunting task. Consequently, the creation of a website like this project aims to resolve these issues, simplifying the decision-making process and streamlining travel planning. By offering personalized itineraries based on user preferences and optimizing routes, it promises to reduce confusion and provide travellers with an effortless and enjoyable journey, thus enhancing the overall travel experience.

1.6 PRODUCT DELIVERY MODE

This project aims to be deployed as a web application with a dedicated frontend and backend for user interactions. The overall objective is to introduce a website for optimized travel planning by creating a user-friendly interface wherein given the user inputs, a travel itinerary is generated. The proposed web application contains main modules such as a home page and a login page. Once logged in, the user will be able to input details of the trip such as place of interest, day of visit, type of interest and arrival date. Further, two additional modules namely, ‘explore’ and ‘recommend’ will help users explore top attractions and other details and recommend the places they have visited. This will be updated in the database and so the recommendation system will have continuously updated information.

1.7 OVERVIEW OF THE REPORT

The following report consists of System Requirement Specifications, System Design Phase and Implementation details of the web application. It highlights all the necessary modules included in the projects and various features that are included in it. The report also includes the software and hardware requirements of the applications.

2. SYSTEM REQUIREMENT SPECIFICATION

2.1 PROBLEM DEFINITION

Using machine learning techniques instead of traditional methods offers a transformative paradigm shift in various industries. Machine learning leverages the power of data and algorithms to extract meaningful insights and make predictions, enabling businesses and organizations to operate more efficiently and effectively.

Machine learning techniques have revolutionized the tourism industry, offering a myriad of benefits that have a profound impact on both businesses and travellers. One of the primary advantages is the ability to personalize the travel experience, enabling businesses to recommend tailored itineraries, accommodations, and activities based on individual preferences and past behaviour. This not only enhances customer satisfaction but also drives customer loyalty and repeat bookings.

This project is aimed at automating the travel planning process, helping users in planning their travel in an efficient way. The objective is to build a travel itinerary web application based on user input exclusively for South India.

2.2 LITERATURE REVIEW

Travel and route planning is a hot-discussed topic due to the growing demand for efficient, sustainable, and personalized transportation solutions. With the integration of technology and data-driven tools, people are increasingly looking for real-time updates, optimal routes, and eco-friendly alternatives to enhance their travel experiences.

A study conducted [1], addresses the common problem of route planning from multiple locations and highlights the low level of confidence among users in finding the optimal route from these locations. The research utilizes genetic algorithms to help plan routes from several locations, incorporating components such as chromosomes, fitness values, selection, crossover, and mutations. The paper introduces the use of the Google Maps API as a data source, providing maps, directions, and distance matrices for route optimization. The research incorporates the element of flexibility in selecting the starting point, which was previously constrained in prior research endeavours.

The results of the research [1] demonstrated that the genetic algorithm approach achieved notable improvements in route optimization, particularly in scenarios involving multiple waypoints or destinations. The algorithm effectively minimized travel distances and reduced the overall time required for reaching all the destinations. The research demonstrated the effectiveness of using genetic algorithms and the Google Maps API for route optimization, providing improved results and assisting in decision-making processes.

Another paper [2] proposes a tourism route selection model based on an improved genetic algorithm (IGA) to find the best tourist route based on the needs of different families and the goals of sustainable tourism development. The proposed IGA addresses the problems of premature convergence and poor local search ability in traditional genetic algorithms (GA) by introducing the ant colony algorithm (ACA) for parameter initialization, an adaptive strategy for adjusting the crossover probability, and the 2-opt optimization algorithm for improving solution quality. The experimental analysis confirms the effectiveness of the proposed method in finding optimal tourist routes while minimizing cost and time for tourists. The paper [2] also discusses the importance of considering cost and time constraints in evaluating the quality of tourist route planning and proposes a formula for evaluating the value of tourists; itinerary experience based on these constraints. The contributions of this paper include the development of a tourism route selection model using IGA, addressing the limitations of traditional GA, and providing a framework for evaluating the quality of tourist route planning.

An improved k-means cluster analysis algorithm is used in paper [3] that shows better clustering results compared to the traditional algorithm. The improved algorithm is applied to the design of an intelligent tourism route planning scheme that considers the tourists; preference metric. The experimental results show that the proposed scheme is reasonable and valuable for both tourism agencies and tourists. The paper introduces a method for determining the initial clustering centres in the k-means cluster analysis algorithm, reducing the dependence on initial values and improving the quality of clustering results. The paper emphasizes the effectiveness of tourism route planning solutions using clustering algorithms, particularly in the context of challenges faced by the tourism industry under the epidemic. The paper highlights the incorporation of wisdom into tourism route planning by using cluster analysis algorithms, aiming to make the planning process more human-like in thinking.

The paper [4] uses a multi-threaded performance test to evaluate the route search submodule of the travel planning system based on artificial intelligence. It tests the tendency of the vertical

search crawler to crawl webpages as the number of multi-threads increases. The paper also conducts a stress test to measure the system's performance changes with an increase in the number of concurrent users. It focuses on the page response latency in the client response performance as the main indicator and performs a load test using virtual users to simulate system load and operating pressure generated by actual users.

Additionally, the paper [4] utilizes association rule mining, also known as shopping basket analysis, to find internal connections between database items and improve the decision-support ability of the application system. However, he provided sources do not explicitly mention the methods used for the overall design and detailed design of the system or the realization of the system prototype.

[5] proposes an artificial intelligence-based wireless sensor travel route planning study, which aims to design reasonable travel routes with the least time cost and the highest experience index. The paper addresses three main problems: classification of influence factors for the travel experience index, allocation of weights using a particle group algorithm-based projection tracing method, and the establishment of an optimization model for the highest tourism experience index. The paper presents the ND-GA algorithm, an improved version of the traditional genetic algorithm, to solve the mathematical model of energy consumption for wireless transmission in travel route planning. The algorithm is used to compute the improved TSP model and obtain 24 travel routes, with the satisfaction time being the highest, allowing visitors to visit all 2015A scenic spots.

The paper provides a detailed analysis of one of these trips, demonstrating the feasibility of the algorithm in designing travel routes. The paper [5] also highlights the potential of AI wireless sensor technology in improving the efficiency of tourism activities, such as using a face recognition system to speed up ticket checking. The application of AI technology in tourism activities is acknowledged, but the paper emphasizes the need for continuous optimization and innovation in AI wireless sensor technology to better serve tourists.

In [6], a travel recommender system is built that uses sentiment analysis, user preferences, and previous reviews to recommend top places to visit. The system calculates the similarity between user preferences and reviews using sentimental attributes (positive and negative) to match the similarity. The system also considers the number of days to be visited and suggests esteemed places based on sentiment polarity and similarity value. The Jaccard similarity is used

to measure the similarity between user preferences and previous reviews. The system prioritizes showing users the items they are most likely to like first on the list.

2.3 BLOCK DIAGRAM

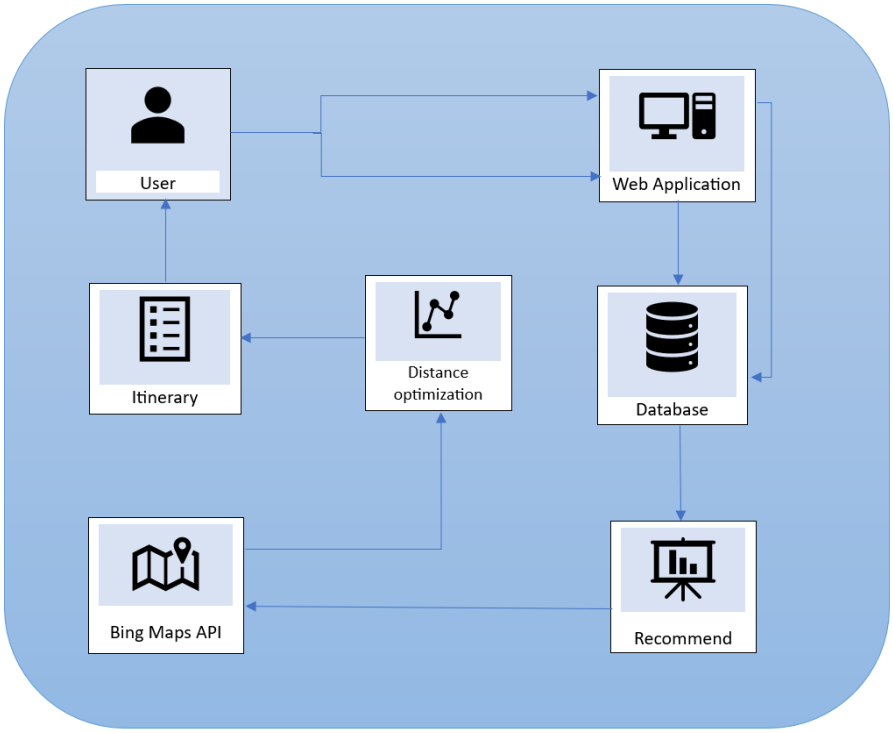


Fig 2.1 Block Diagram

2.4 SYSTEM REQUIREMENTS

2.4.1 USER CHARACTERISTICS

The users need to have a stable internet connection as the itinerary generating system will be delivered as a web application. The application should provide the user with the ability to log in either as a customer or a stakeholder of the retail store. The system should be able to identify if the login was performed by a customer or stakeholder and direct to web pages based on the user.

- If the login is performed by a customer:
    i. The system should be able to capture the requirements of the users such as place of interest, number of days of visit, start date and type of interest.

ii. Based on the inputs, the system should filter the attractions in the place and recommend the top n attractions.

iii. The system should then optimize the distance between these attractions and generate a travel itinerary.

iv. If the customer wants to provide feedback about any destination, he/she had been, then it can be done in the recommend module of the web application.

- If the login is performed by a business stakeholder:

  The system should display the updated destination information to the stakeholder.

I. Home Page:

II. Recommend:

III. Explore:

IV. About Us

2.4.2 SOFTWARE AND HARDWARE REQUIREMENTS

Table 2.1 Hardware Requirements

Hardware Requirements	**Configuration**
Server Hardware	
CPU	12th Gen Intel(R) Core (TM) i5-12500H 2.50 GHz
RAM	8.0 GB or more
Storage	500 GB
Client Devices	
Computers/Laptop	Windows - capable of running modern browsers

Table 2.2 Software Requirements

Software Requirements	**Configuration**
Operating System	Microsoft Windows 11 Home Single Language

Integrated Development Environment (IDE)	Google Collaboratory, Jupyter Notebook, Spyder
Front-End Development a. Front-End Framework	HTML
b. Styling Framework	Bootstrap, CSS, JavaScript
Back-End Development	
c. Back-End Framework	Django, Python
d. Database management System	SQL
API's	Bing Maps API
External Components	Travel destination website (https://www.nativeplanet.com/tourist-places/), Latitudes and longitudes generated from Geocode extension in Google Sheets
Algorithms	Heuristics
NLP frameworks	NLTK (Natural Language Toolkit)

2.4.3 CONSTRAINTS

The users should have access to a good internet connection, along with properly working electronic devices to use the web application. To get the travel itinerary, the users should necessarily input all the fields in the module for travel itinerary generation.

2.5 DATASET DESCRIPTION

The dataset used for this project for the development of the proposed travel itinerary generation and recommendation system was extracted from travel websites using different advanced web scraping techniques. These datasets are fundamental components that empower the software to offer travel planning and destination recommendations. The dataset collection comprises two essential components: the travel destination dataset and the travel destination review dataset. Each dataset has a distinct role in shaping the functionality and effectiveness of the software, focusing on distance optimization and personalized recommendations.

2.5.1 TRAVEL DESTINATION DATASET

The travel destination dataset is a cornerstone element in the creation of travel itineraries that are optimized for distance and efficiency. Leveraging the synergy of Selenium web scraping and Power Automation, a comprehensive repository of travel information is assembled. The dataset is structured in JSON format and consists of the following essential variables:

i. **State:** The geographical state where the travel destination is situated.
ii. **City:** The specific city within the state where the travel destination is located.
iii. **Destinations:** A comprehensive list of notable landmarks, attractions, or places of interest within the city.
iv. **Best Season to Visit:** Information about the most suitable time of the year to visit each destination, considering weather and local events.
v. **Famous For:** A description highlighting the type of destinations that the city is famous for.

2.5.2 TRAVEL DESTINATION REVIEW DATASET

The travel destination review dataset enriches the software's capacity to offer personalized travel destination recommendations to users. This dataset draws from user-generated reviews and feedback, contributing to the software's ability to tailor recommendations based on individual preferences and reviews. The dataset encompasses:

i. **User ID:** A unique identifier assigned to each user providing a review.
ii. **Destination ID:** A reference number linking the review to a specific travel destination.
iii. **Review Text:** The textual content of the user's review encompasses personal experiences, opinions, and recommendations.
iv. **Rating**: A numeric value assigned by the user to denote their level of satisfaction with the respective travel destination.
v. **Date of Review:** The date on which the review was submitted by the user.

2.5.3 DATA SOURCES

- The dataset for building the travel itinerary generative system was extracted from travel websites using different advanced web scraping techniques. The main web scraping tool used is the Selenium web scrape. Selenium is used to extract all the destinations in India that were listed on the website https://www.nativeplanet.com/tourist-places/ by

automating the web scrape. The web-scraped data was then collected in a JSON format and saved in a text file.

- Travel Destination Review Dataset is extracted using selenium and power automation tool. This data helps to recommend travel destinations, and this can be used for distance optimization.
- The geospatial location is obtained using an extension in Google Sheets namely, Geo Encode and the distance matrix between the destinations will be calculated and retrieved using Bing Maps API.

2.5.4 DEPENDENCIES ON DATASETS

The accuracy of the travel itinerary generation and recommendation system heavily depends on the quality and freshness of the travel destination dataset. Any updates or inaccuracies in this dataset can affect the generated itineraries and recommendations.

3. SYSTEM DESIGN PHASE

3.1 SYSTEM ARCHITECTURE

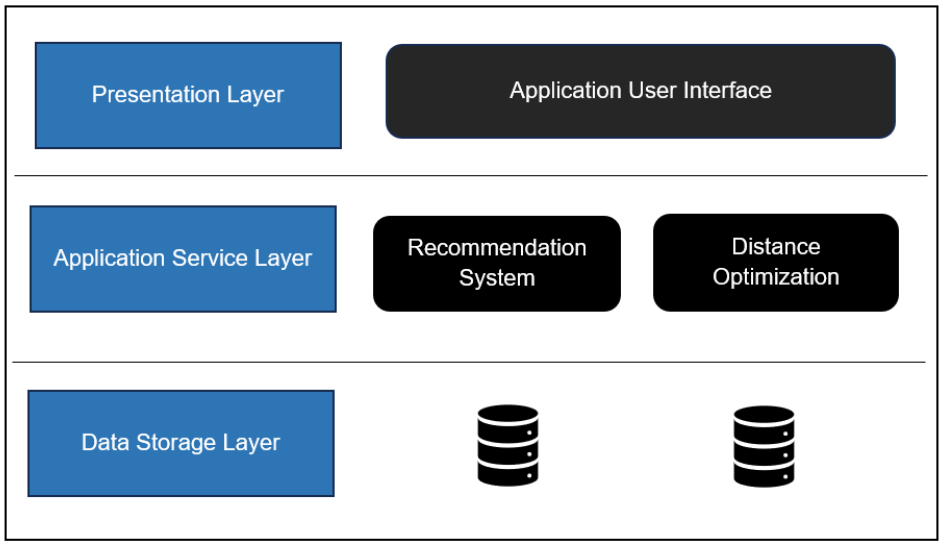


Fig 3.1 System Architecture

The system follows a three-tier architecture where the user sends a request to the server which in turn is connected to the database. Information from the database is sent and received by the server, based on requirement.

Presentation Layer:

This layer is the topmost layer, which is the user interface layer. The user interacts with this layer of the system. This layer comprises the following:

- Registration and login (for customer and business stakeholder)
- Gives itinerary (for customers)
- Takes recommendation and feedback (from customers)
- View and manage database (for retailers/business stakeholder)

Application Service Layer:

The primary function of application service layer is to provide specific services to the user. The main functionality within this layer will be the recommender system which will recommend the top n attractions, optimize the distances and provide the travel itinerary to the customer.

Data Storage Layer: This layer consists of all the databases that will be relevant to the web application. There will be three two databases stored in this layer.

3.2 MODULE DESIGN

The web application is divided into the following modules, as described below:

1. Registration and Login: The system will have a registration and login module by which both the user and stakeholder can sign up or login into the application. New users can register and then proceed to use the application whereas an already registered user can login to use the application.
2. Home: This is where the user will have to fill in the required fields for itinerary generation. Once the user inputs details such as 'place of interest', 'days of visit', 'type of interest' and date of arrival, the travel itinerary will be shown on the same page. The itinerary will contain the places to visit in order, along with a small description about the place.
3. Recommend: This module is where the users can provide feedback and rating about any destination they visited. This will help in updating of information in the review database in real-time. The user is expected to provide details like 'place', 'city', 'district', 'state', 'review', and 'rating'.
4. Explore: In this module, the user can explore different destination spots in the southern part of India. Here the destination places are displayed with respect to each state. The user can choose the state they want to explore.
5. About us: In this module, The user can understand the aim, and objective of this website. Also, helps to understand the website in a nutshell.

3.3 DATABASE DESIGN

3.3.1 DATA FLOW DIAGRAM

A data flow diagram (or DFD) is the representation of the flow of data in a business process in a graphical form. It provides insight into the inputs and outputs of each entity and helps in visualising the transfer of information. DFD is of different levels and each level is numbered as 0, 1, 2 and so on. With each level, the representation is given in more detail. For example, level 0 provides an abstraction view, showing the whole system as a single process with its relationship to external entities whereas level 1 decomposes the level 0 diagram into multiple

processes and breaks down the high-level processes of level 0 into subprocesses. Level 1 DFD highlights the main function of the system. The detail increases with each level of the DFD. In this document, DFD up to level 2 will be represented.

DFD LEVEL 0

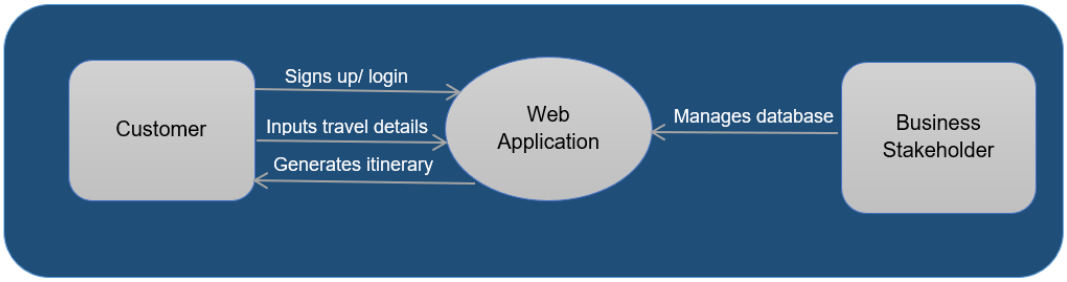


Fig 3.2 DFD Level 0

DFD LEVEL 1

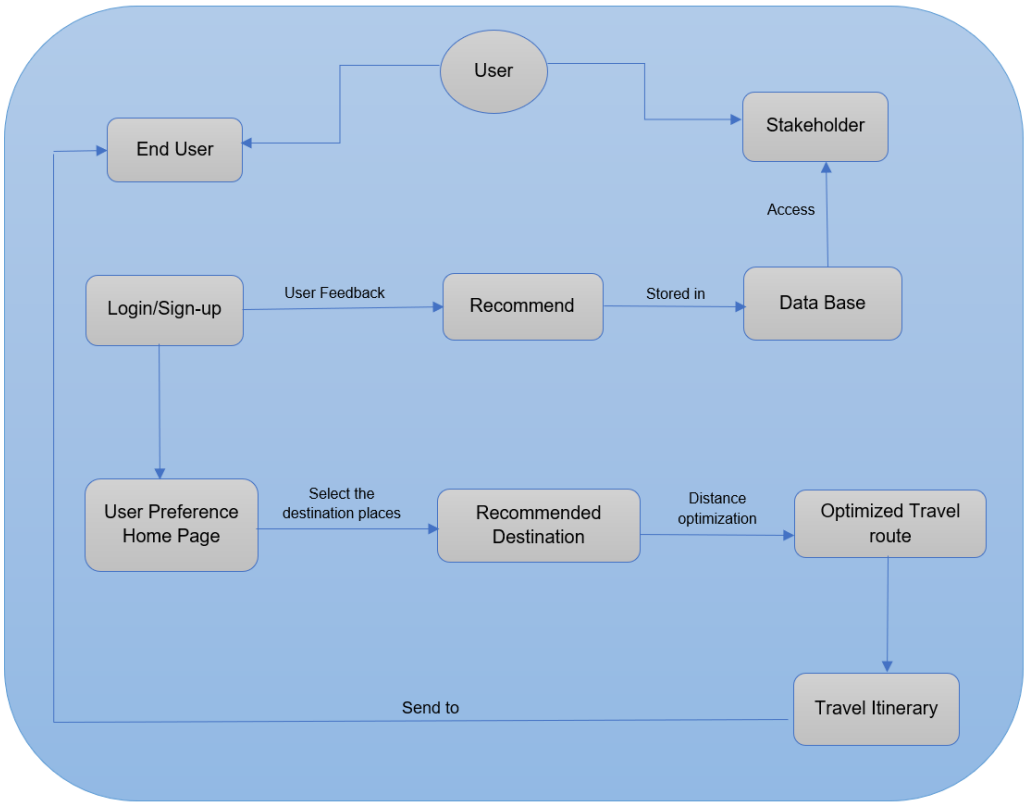


Fig 3.3 DFD Level 1

DFD LEVEL 2

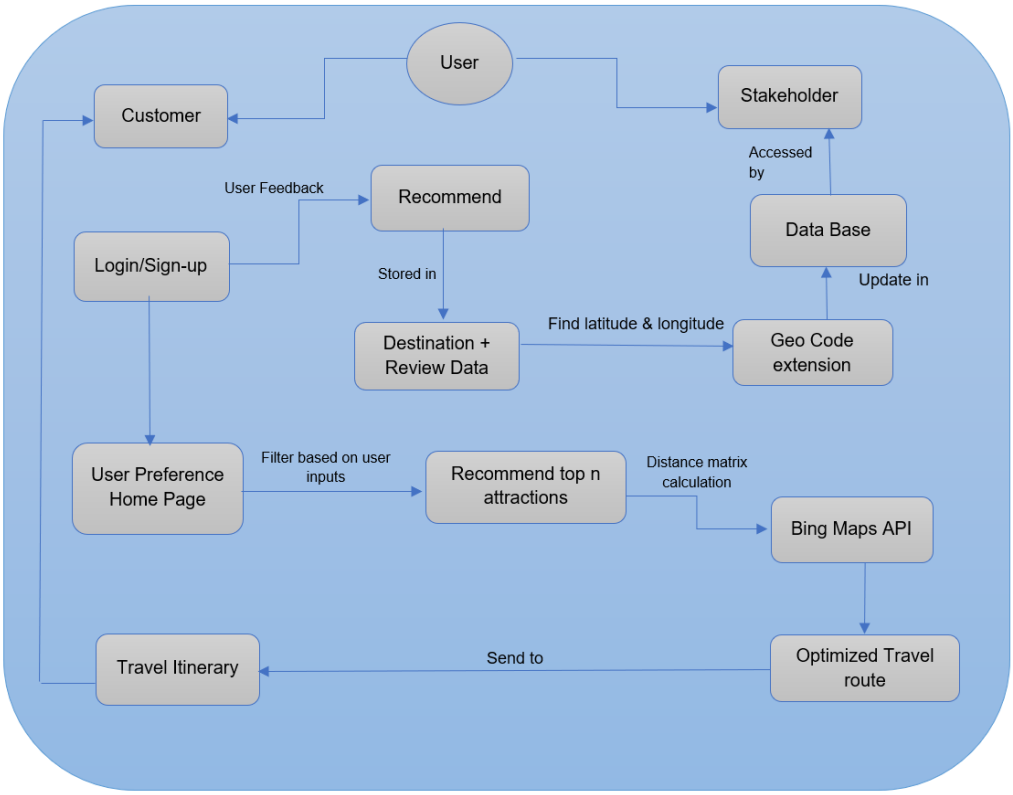


Fig 3.4 DFD Level 2

3.3.2 ERD

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.

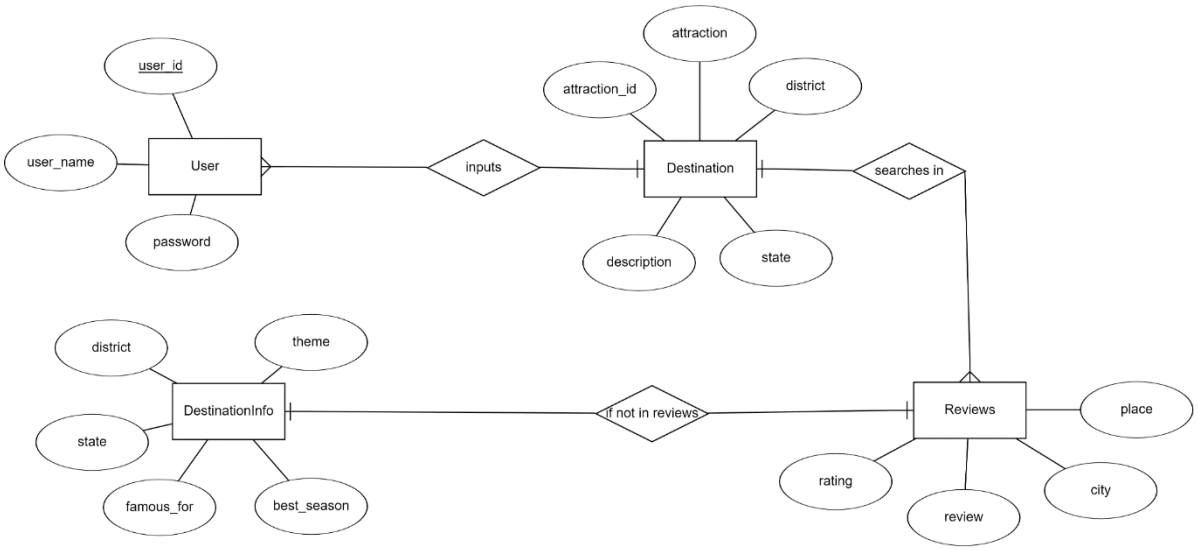


Fig 3.5 ERD

3.4 INTERFACE DESIGN

Home Page before login/sign up:

On accessing the website, the user would be able to see the web page as shown above. The webpage contains a main bar which displays the different sections or features that are available on the website. Features like Home, Recommend, Explore, About Us. Also, to access those features the users must either login or signup to the application. To achieve that the index page or the landing page contains the login and signup button so that the user can login to the application.

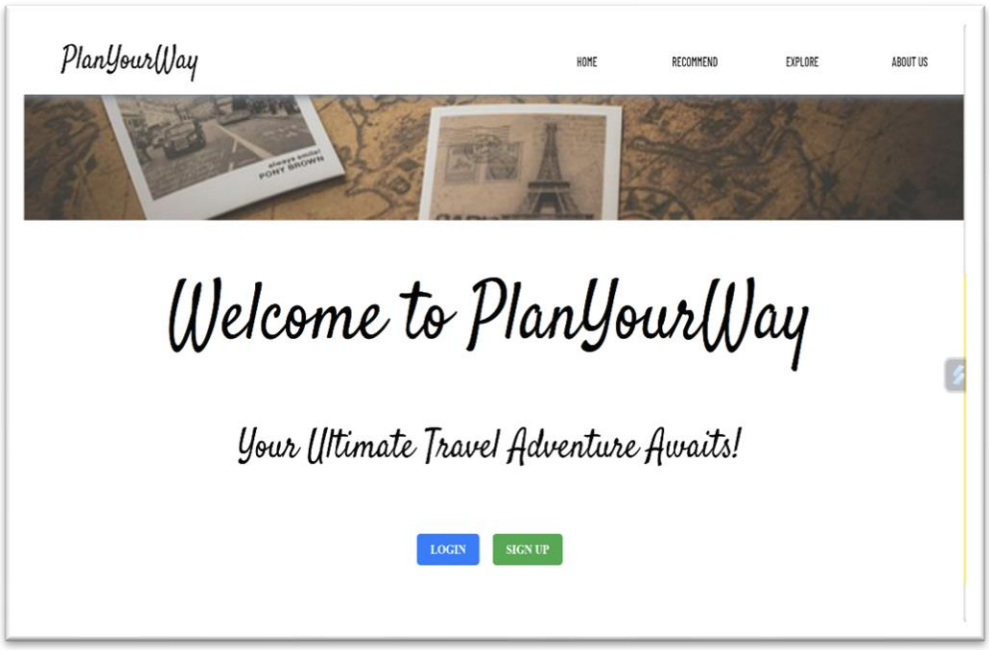


Fig 3.6 Landing Page

Signup:

On clicking on the Signup, the user will be navigated to Signup page. Here, the user must provide details like username, firstname, lastname, email ID, password. Once the user signup to the application, the user will be navigated to the login page.

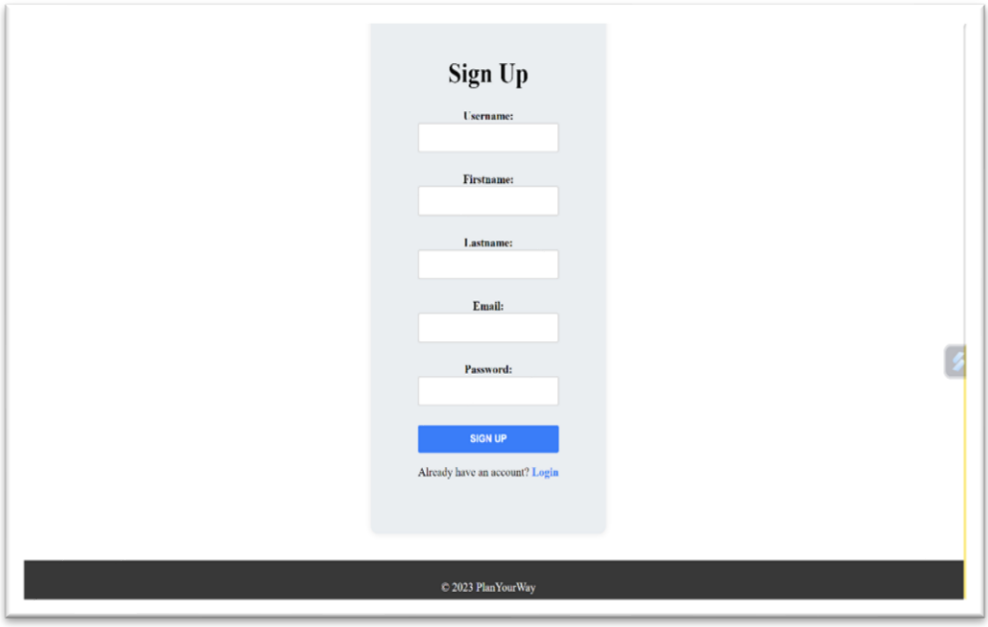


Fig 3.7 Signup Page

Login:

In the login page, the user must provide the username and password to access the features of application.

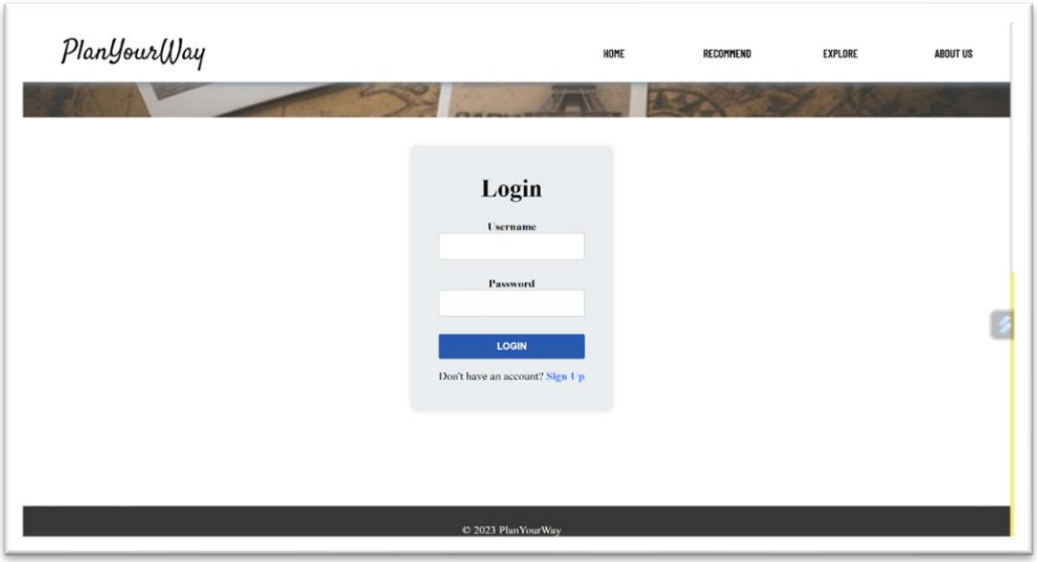
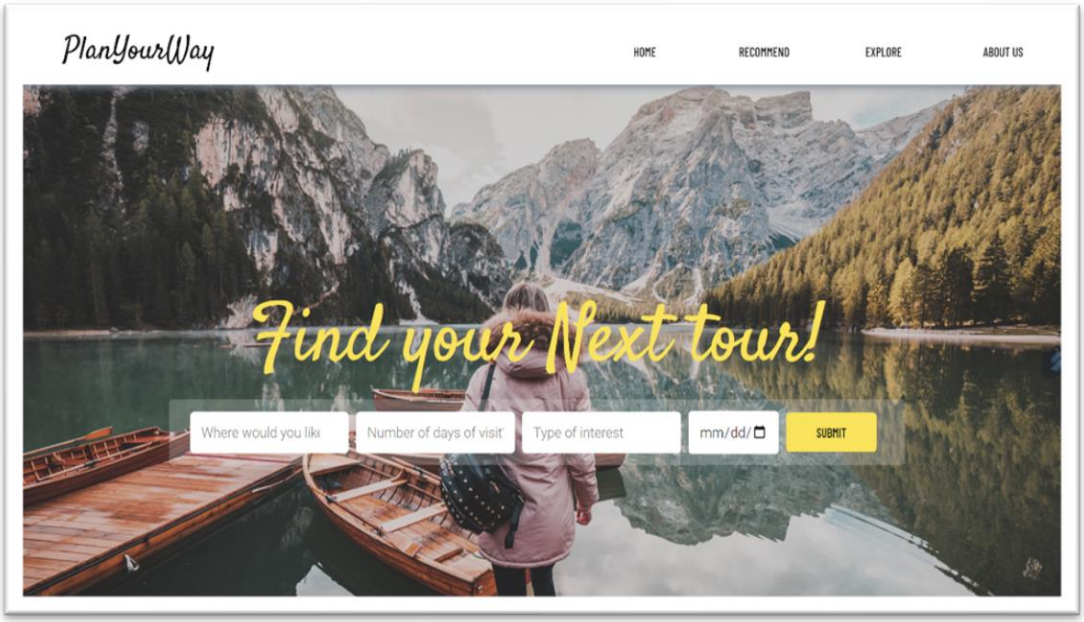


Fig 3.8 Login Page

Home Page after Login:

After successful login the user will land in the home page where the user must provide the details of the like where the user is intended to go i.e., the destination state, no of days, type of interest, Tentative date of visit which is optional to generate the travel itinerary. Once provided with the details and submitted the same the user will be navigated to the itinerary/result page.



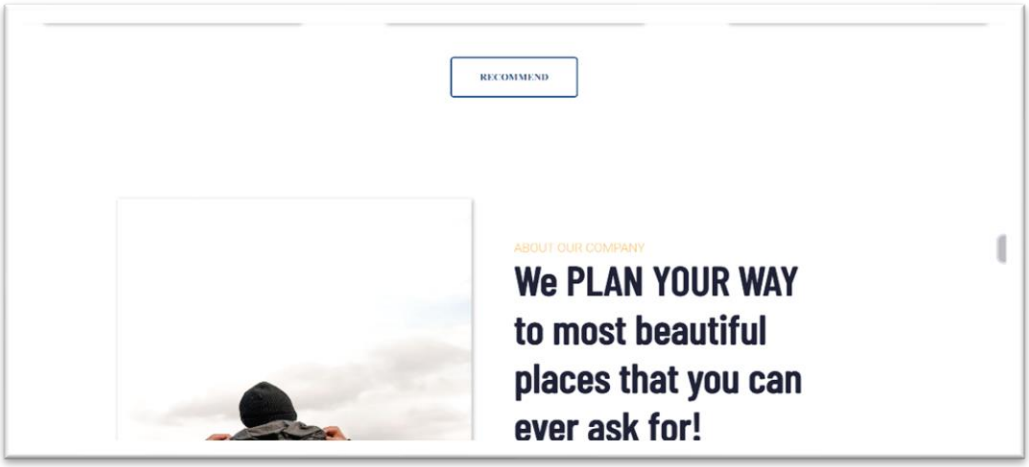


Fig 3.9 Home Page

Result/Itinerary page:

The user will be navigated to the resultant page only after providing necessary inputs in the home page. In the resultant page will display the travel itinerary to the user based on the user preferences. The travel Itinerary will provide the destination places to be visited by the user while they place to visit that state or city.

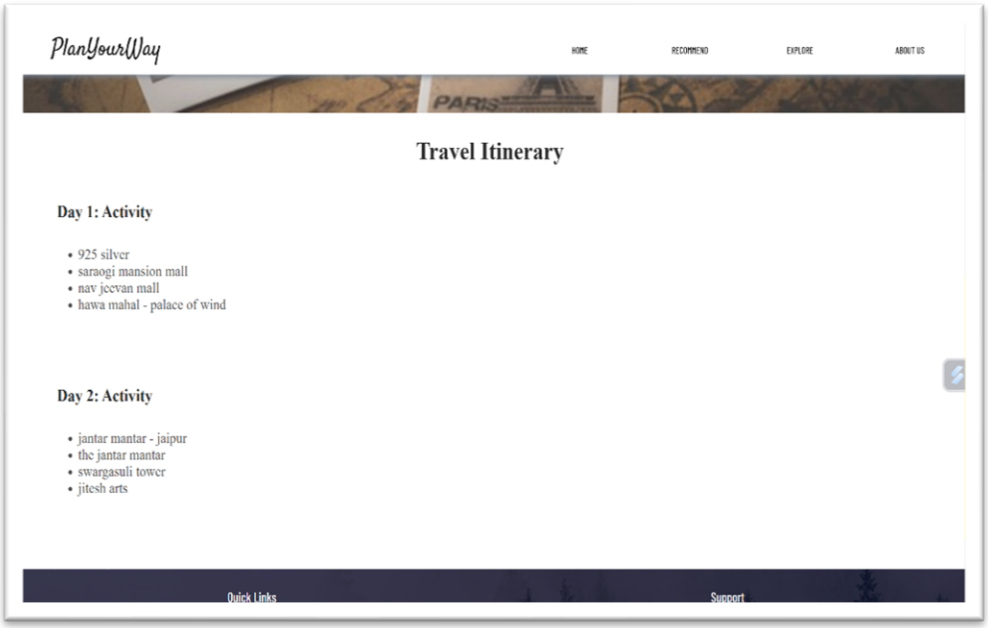


Fig 3.10 Itinerary Page

Recommend page:

The user can navigate to the recommend page using the recommend button both in the main bar and home page. Recommend page contains a form where the user can provide the review about any of the destination places, they have visited. These reviews added by the user can help in providing better recommendation of destinations in future. This page also contains the contact details of the company running the website.

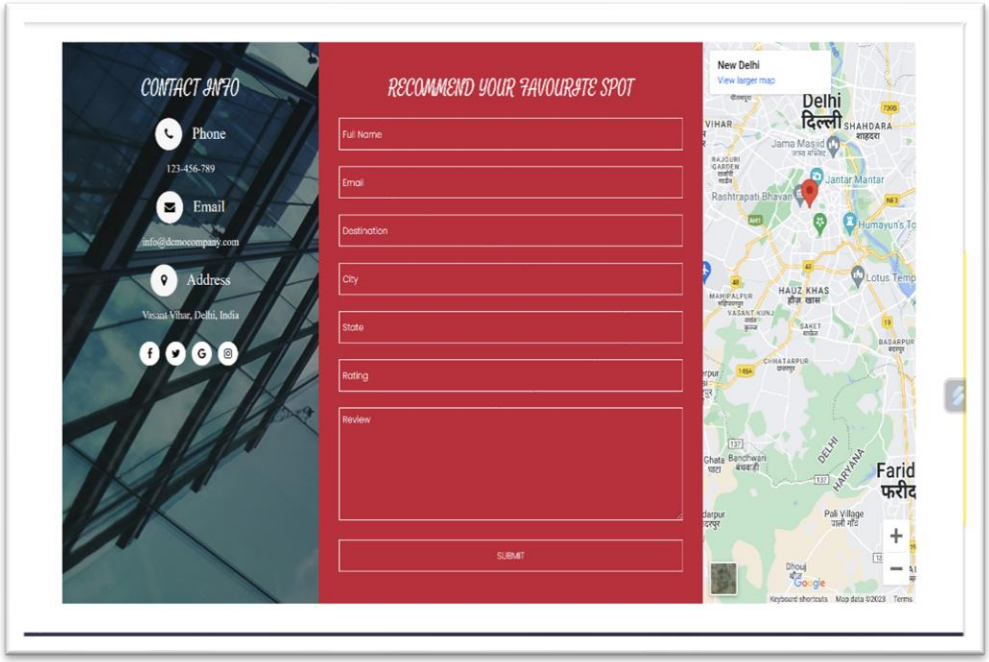


Fig 3.11 Recommend Page

Explore Page:

The Explore Page showcases India's top destinations, with a visually appealing design. It features a prominent header, a search bar, and filters for sorting by type. A captivating carousel highlights the top places, and scrollable lists present destinations categorized by type. Each entry includes an image, and name, enabling users to explore top and attractive spots.

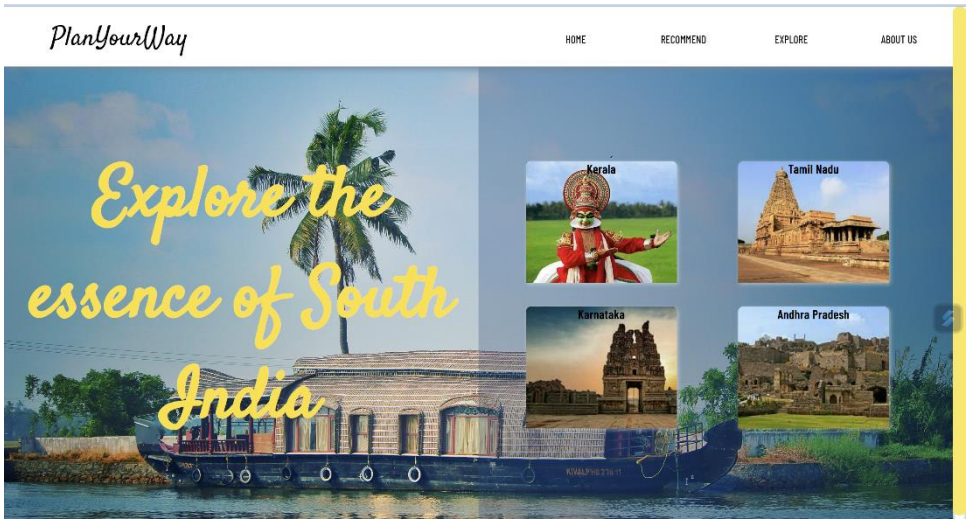


Fig 3.12 Explore Page

About Us:

About Us page provides insights into the platform's creators and mission. It typically features details about the team's travel expertise, the site's commitment to helping travelers. This page fosters trust connects with visitors and conveys the website's dedication to enriching travel experiences.

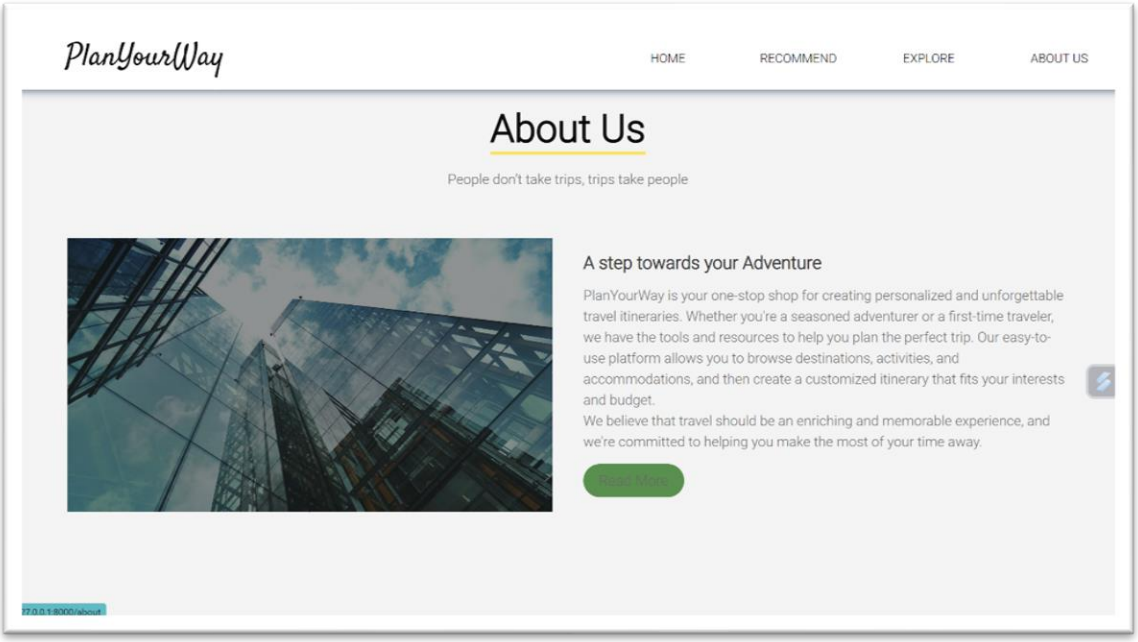


Fig 3.13 About Us Page

4. IMPLEMENTATION

4.1. IMPLEMENTATION APPROACHES

Implementation of the travel itinerary website was divided into sub-tasks which were accomplished in the following order:

1. Creating the frontend for the website

The implementation of the frontend is done entirely with the help of the Django framework, along with basic HTML and CSS.

- **Sign-up and login page:** A new user must sign up to the website to generate the travel itinerary according to his/her preference. The sign-up page takes in information such as the username, name, email and password from the user. An already existing user can login using their username and password. For the creation of the sign-up page, the inbuilt UserCreationForm of Django was used. Similarly, for the authentication of users at the time of login, the in-built is authenticated function under Authentication Form of Django is used.
- **Homepage:** The homepage displays a form where the user must input his/her travel preferences. The user inputs are loaded into the website by connecting the travel destination dataset to an SQL server, which in turn is connected to the website through the Django framework. To generate the itinerary based on the user input destinations, recommended destinations will be first iterated based on state, city, interest. These POST request details will be provided as a GET request to the resultant page.
- **Travel Itinerary Generation Page**: This is the resultant page where an AI-generated travel itinerary will be generated based on the user's preference. Here, the details entered by the user will be retrieved as a GET request and first destination recommendation will be provided based on the input and by applying geo location. Later, these retrieved top n recommended destinations will undergo distance optimization in order to save time and enjoy the visit.
- **Recommend Page:** The recommend page contains a form where the user can provide the review about any of the destination places, they have visited. The provided review details will be saved into a database created for storing the same. These reviews added by the user can help in providing better recommendation of destinations in future.

2. Building the recommendation system to retrieve top n attractions

- The data come from four different files. This is first transformed and integrated into a single dataset suitable for the analysis. The destination dataset and review dataset are combined using Fuzzy matching, after which the geographical coordinates are extracted using the Geocode extension in Google Sheets.
- Sentiment Analysis is employed on the reviews to get the sentiment and obtain a ratio of positive to total reviews ratio. This will help in understanding the popularity of a destination.
- Natural Language Processing techniques is used to find the tourism type of each destination. Based on this, six types - 'Adventure', 'Pilgrimage', 'Heritage', 'Recreation', 'Environmental' and 'Others' are formed.
- The recommendation system used here is not based on machine learning techniques. This is more of Ranking and condition-based search, based on which the destinations will be retrieved.
- First, the top-rated destinations with high ratio are filtered. Then 50 % of these filtered destinations will be chosen based on the type of interest specified by the user. The system will then take destinations from other types. This is done so that the itinerary will give higher importance to destinations belonging to the type of interest that the user specified, at the same time will not be entirely of those type. For instance, for a user who gave 'Heritage' as input might not just want to see palaces and monuments throughout his stay in that destination. Thus, doing this the system makes sure that there is a combination of all types.

3. Distance Optimization

The solve_tsp_simulated_annealing function from the python_tsp library implements the simulated annealing algorithm to find an approximate solution to the Traveling Salesman Problem (TSP). The TSP is a classic optimization problem where the goal is to determine the most efficient route that visits a set of destinations exactly once and returns to the starting point. Simulated annealing is a probabilistic optimization algorithm inspired by the annealing process in metallurgy.

The algorithm starts with an initial solution, which represents a possible route in the TSP context. It then iteratively explores neighboring solutions by making small changes to the current route. These changes can involve swapping, inserting, or reversing the order of visited

destinations. Simulated annealing introduces a temperature parameter that controls the probability of accepting worse solutions early in the search. As the algorithm progresses, the temperature decreases, and the likelihood of accepting worse solutions diminishes, allowing the algorithm to converge towards an optimal or near-optimal solution.

4.2 CODING STANDARD

As discussed earlier, Django, HTML and CSS is leveraged for the frontend, python is used for building the recommendation system and finally, SQL is used in the backend for the application. The web application is coded in such a way that the user can input the values and change the output for personalisation according to their desire.

1. Naming Conventions

All classes, functions, variables and constants have been named such that they are self-explanatory. Given below are the naming conventions that have been used throughout the code:

- **Classes**: All class names begin with upper case letters and are followed by lower case letters.
- **Functions**: All function names are written entirely in lower case letters, with underscore separating function names having more than one word.
- **Variables**: Variable names follow the same conventions as functions.
- **Constants**: Constants are defined entirely in upper case letters with underscores separating more than one word.
- **With respect to SQL:**
    (i) Reserved keywords have been written in upper case letters.
    (ii) Table names start with uppercase letters.
    (iii) All column names have been written in lowercase letters, with underscore separating words.

2. Indentation

Standard python indentation of one tab has been used.

3. Variable Annotation

Annotation for module level variables, class and instance variables and local variables have a single space after the colon. There is no space left before the colon. If an assignment has a right hand side, then the equality sign has exactly one space on both sides of the sign.

4. Comments

- Comments are given before the corresponding line of code rather than the same line of code, in order to increase readability.
- Indentation of the comments are at the same indentation depth as of the code.
- Comments are preceded by a blank line and are immediately followed by the code they refer to.
- There is always a space left between the '#' symbol and the starting of the comment line.
- Comments have always been begun with upper case letters, except in cases where the line may be a continuation of a sentence or the term is a code(s) and is case-sensitive.
- Commented code has been marked as "Code intentionally commented" to prevent accidental removal.

4.3 CODING DETAILS

There are several python libraries which are used to implement the analysis code for the application. Some of these libraries include pandas, numpy, geopy etc. Since the main aspect of the project is to display the homepage and give recommendations to the users, snippets of code for these are given in the section below.

4.4 CODE

The codes for views.py and html pages are attached as they form the most important part of the project. Therefore, the codes for the same is given below.

(a) Views.py

```
from django.shortcuts import HttpResponse,render,redirect
from django.contrib.auth.models import User
from django.contrib.auth import authenticate,login
```

```
from django.contrib import messages
from .models import Traveldata, reviewdata


import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
from geopy.distance import great_circle
from joblib import load
from python_tsp.exact import solve_tsp_dynamic_programming
from python_tsp.heuristics import solve_tsp_simulated_annealing
import requests


def index(request):
    return render(request,'index.html')


def signupdetails(request):
    if request.method=='POST':
        uname=request.POST['username']
        fname=request.POST['fname']
        lname=request.POST['lname']
        email=request.POST['email']
        password=request.POST['password']


        if User.objects.filter(email=email).exists():
            messages.error(request,'email already exists, try login')
            return redirect('/signupdetails')
        elif User.objects.filter(username=uname).exists():
            messages.error(request,'username already exists, try login')
            return redirect('/signupdetails')

        else:
            myuser=User.objects.create_user(uname,email,password)
            myuser.first_name =fname
            myuser.last_name =lname

            myuser.save()

            messages.success(request,'Your Account has been successfully created!')



            print(uname,fname,lname,email,password)
        return redirect('/logindetails')

    return render(request,'signupdetails.html')
```

```
def home(request):
    if request.method=='POST':
        state =request.POST['state']
        city =request.POST['City']
        days=request.POST['days']
        type=request.POST['typeofInterest']
        print(state,days,city,type)
        # Store the values in the session
    return render(request,'home.html')



def result(request):

    state =request.GET['state']
    city = request.GET['City']
    n=int(request.GET['days'])
    interest=request.GET['typeofInterest']


    df3 =
            pd.read_excel(r"C:/Users/asus/VScode/Project3/webproject/traveldest/savedmodels/d
            ata_south_updated_v2.xlsx")

    with open('C:/Users/asus/VScode/Project/websitedemo/travel/savedmodels/BingMap.txt',
            'r') as f:

        api_key = f.read()



    state_filtered = df3[df3['State'] == state]
    dis_filtered = df3[df3['City'] == city]


    def recommend_destinations(latitude, longitude, place_df):
        if place_df.shape[0] != 0:
            merged = state_filtered.merge(place_df, how = 'left', indicator = 'True')
            merged = merged[merged['True'] == 'left_only']
            merged.drop(['True'], axis = 1, inplace = True)
            merged['distance_to_location'] = merged.apply(
                lambda row: great_circle((latitude, longitude), (row['Latitude'],
            row['Longitude'])).miles, axis=1
            )
# Calculate distances to destinations within the state
            recommended_destinations = merged.sort_values(by='distance_to_location')
 # Sort destinations by distance and get the top N recommendations
```

```
            recommended_destinations =
            recommended_destinations[recommended_destinations['distance_to_location'] < 250]
            return recommended_destinations
        else:
            state_filtered['distance_to_location'] = state_filtered.apply(
                lambda row: great_circle((latitude, longitude), (row['Latitude'],
            row['Longitude'])).miles, axis=1
            )
            recommended_destinations = state_filtered.sort_values(by='distance_to_location')
# Sort destinations by distance and get the top N recommendations
            recommended_destinations =
            recommended_destinations[recommended_destinations['distance_to_location'] < 250]
            return recommended_destinations

    top_filter = dis_filtered.sort_values(by=['Rating', 'Ratio'],
            ascending=False).reset_index(drop=True)

    place_df = pd.DataFrame()

    if interest in top_filter['Type'].tolist():

        interest_filter = top_filter[top_filter['Type'] == interest].sort_values(by=['Rating',
            'Ratio'], ascending=False).reset_index(drop=True)

        number = n * 4

        place_df = pd.concat([place_df, interest_filter.head(int(0.5 * number))],
            ignore_index=True)

        number -= place_df.shape[0]

        place_df = pd.concat([place_df, top_filter[top_filter['Type'] != interest].head(number)],
            ignore_index=True)

        if place_df.shape[0] != n * 4:

            remaining = n * 4 - place_df.shape[0]

            dest1_lat, dest1_longi = place_df['Latitude'][0], place_df['Longitude'][0]

            recommendations = recommend_destinations(dest1_lat, dest1_longi, place_df)

            keys = recommendations['Type'].unique()

            track_dict = {key: 0 for key in keys}

            while place_df.shape[0] != 4 * n:

                for key, value in track_dict.items():

                    place_df = pd.concat([place_df, recommendations[recommendations['Type'] ==
            key].head(1)], ignore_index=True)

                    track_dict[key] = value + 1

                    if place_df.shape[0] == 4 * n:

                        break
```

```
        final_df = place_df
    else:
        dest1_lat, dest1_longi = top_filter['Latitude'][0], top_filter['Longitude'][0]
        recommendations = recommend_destinations(dest1_lat, dest1_longi, place_df)
        final_df = recommendations.groupby('Type').apply(lambda x:
            x.head(n)).sort_values(by=['distance_to_location', 'Rating',
            'Ratio']).reset_index(drop=True)
        final_df.head(n * 4)



    coordinates = final_df[['Latitude', 'Longitude']].to_records().tolist()
    api_key = api_key
    travel_mode = "driving"
# You can change this to "walking," "transit," etc. as needed
    distance_matrix = []
    for origin in coordinates:
        row = []
        for destination in coordinates:
            if origin == destination:
                row.append(0.0)
# Distance from a location to itself is 0
            else:
                origin_str = f"{origin[1]},{origin[2]}"
# Construct the API request for each pair of coordinates
                destination_str = f"{destination[1]},{destination[2]}"
                url =
            f"https://dev.virtualearth.net/REST/v1/Routes/DistanceMatrix?key={api_key}&origin
            s={origin_str}&destinations={destination_str}&travelMode={travel_mode}"
                response = requests.get(url)
                data = response.json()
                distance = data["resourceSets"][0]["resources"][0]["results"][0]["travelDistance"]
 # Extract the distance from the API response
                row.append(distance)
        distance_matrix.append(row)
```

```
    for row in distance_matrix:
        print(row)
    permutation2, dismat2 = solve_tsp_simulated_annealing(np.array(distance_matrix))
    final2_data = []
    for i in permutation2:
        place = final_df.iloc[i, 1]
        final2_data.append(place)

    values_per_day = len(final2_data) // n
    result_dict = {}
    for day in range(1, n + 1):
        start_index = (day - 1) * values_per_day
# Calculate the start and end indices for the values
        end_index = day * values_per_day
        day_values = final2_data[start_index:end_index]
 # Slice the values for the current day
        result_dict[f'Day {day}'] = day_values
    return render(request,'result.html',{'result' : result_dict})


def logindetails(request):
    if request.method=='POST':
        username=request.POST['username']
        pass1=request.POST['password']
        user=authenticate(username=username,password=pass1)
        if user is not None:
            login(request,user)
            return render(request,'home.html')
        else:
            messages.error(request,"Bad Credentials! Try Login Again")
            return redirect("/logindetails")
    return render(request,'logindetails.html')
```

```
def recommend(request):
    return render(request,'recommend.html')


def about(request):
    return render(request,'about.html')

def explore(request):
    return render(request,'explore.html')
```

HTML Pages Code for Customer:

- **Home.html**

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>PlanYourWay</title>
  <link rel="stylesheet" type="text/css" href="{% static 'css/style.css'%}">
</head>
<body>

  <!--============Nav Bar==================-->

  <section class="nav-bar">
    <div class="logo">PlanYourWay</div>
    <ul class="menu">
      <li><a href="home">home</a></li>
      <li><a href="recommend">recommend</a></li>
      <li><a href="explore">explore</a></li>
      <li><a href="about">about us</a></li>
    </ul>
    </div>


  </section>



  <!--================Banner=================-->
```

```
 <section class="banner">
  <div class="banner-text-item">
   <div class="banner-heading">
    <h1>Find your next Tour!</h1>
   </div>
   <form class="form" method="GET" action="result">
    <input type="text" list="mylist" placeholder="Where would you like to go?"
name="state" required>
     <datalist id="mylist">
      <option>kerala</option>
      <option>goa</option>
      <option>Karnataka</option>
      <option>andhra pradesh</option>
      <option>tamil nadu</option>
     </datalist>
     <input id="City" placeholder="City" name ="City" required>
     <input id="noofdays" placeholder="Number of days of visit?" name ="days"
required>
     <input id="typeofInterest" placeholder="Type of interest" name="typeofInterest"
required>
     <input type="date" class="date">
     <button type="submit" class="book">Submit</button>
   </form>
  </div>
 </section>

 <!--==========Services===============-->

 <div class="service-text">
  <h2>What makes us Special</h2>
 </div>
 <section class="service">
  <div class="service-item">
   <img src="https://res.cloudinary.com/dxssqb6l8/image/upload/v1605293634/tour-
guide_onzla9.png">
   <h2>Customized Itineraries</h2>
  </div>
  <div class="service-item">
   <img
src="https://res.cloudinary.com/dxssqb6l8/image/upload/v1605293738/reliability_jbp
n4g.png">
   <h2>100% Trusted Tour Experience</h2>
  </div>
  <div class="service-item">
   <img
src="https://res.cloudinary.com/dxssqb6l8/image/upload/v1605293635/experience_a3
fduk.png">
   <h2>Handpicked Experiences</h2>
  </div>
```

```
    <div class="service-item">
      <img
src="https://res.cloudinary.com/dxssqb6l8/image/upload/v1605293634/feedback_s8z7
d9.png">
      <h2>98% Our Travelers are Happy</h2>
    </div>
  </section>

  <!--==============Places==================-->

  <section class="places">
    <div class="places-text">
      <small>FEATURED TOURIST DESTINATIONS</small>
      <h2>Favourite Places</h2>
    </div>

    <div class="cards">
      <div class="card">
        <div class="zoom-img">
          <div class="img-card">
            <img src="https://www.holidify.com/images/bgImages/MANALI.jpg">
          </div>
        </div>

        <div class="text">
          <span
class="rating">&#11088;&#11088;&#11088;&#11088;&#11088;</span>
          <h2>Manali, Himachal Pradesh</h2>
        </div>
      </div>
      <div class="card">
        <div class="zoom-img">
          <div class="img-card">
            <img src="https://www.transindiatravels.com/wp-content/uploads/the-taj-
mahal-agra.jpg">
          </div>
        </div>
        <div class="text">
          <span
class="rating">&#11088;&#11088;&#11088;&#11088;&#11088;</span>
          <h2>Taj Mahal, Agra</h2>
        </div>

      </div>
      <div class="card">
        <div class="zoom-img">
          <div class="img-card">
            <img
src="https://i.pinimg.com/originals/b8/a6/86/b8a6860430b195e089c26df1c5577732.j
pg">
```

```
        </div>
      </div>

      <div class="text">
        <span
class="rating">&#11088;&#11088;&#11088;&#11088;&#11088;</span>
        <h2>Munnar, Kerala</h2>
      </div>
    </div>
    <div class="card">
      <div class="zoom-img">
        <div class="img-card">
          <img
src="https://tse3.mm.bing.net/th?id=OIP.IlQYgL4BR0rNrHxIZ7HK8gHaE6&pid=Api&P=0&h=220">
        </div>
      </div>

      <div class="text">
        <span
class="rating">&#11088;&#11088;&#11088;&#11088;&#11088;</span>
        <h2>Udaipur, Rajasthan</h2>
      </div>

    </div>
    <div class="card">
      <div class="zoom-img">
        <div class="img-card">
          <img
src="https://tse1.mm.bing.net/th?id=OIP.7UJ5mwXP71Lw7_JErAgWHAHaFj&pid=Api&P=0&h=220">
        </div>
      </div>

      <div class="text">
        <span
class="rating">&#11088;&#11088;&#11088;&#11088;&#11088;</span>
        <h2>Cherrapunji, Meghalaya</h2>
      </div>

    </div>
    <div class="card">
      <div class="zoom-img">
        <div class="img-card">
          <img
src="https://media2.thrillophilia.com/images/photos/000/039/872/original/1607412230_shutterstock_1301320006.jpg?w=753&h=450&dpr=1.5 2x">
        </div>
      </div>
```

```
        <div class="text">
          <span
class="rating">&#11088;&#11088;&#11088;&#11088;&#11088;</span>
          <h2>Baga Beach, Goa</h2>
        </div>
      </div>
    </div>
    <div class = "recommend-text">
    <a href="recommend">RECOMMEND</a>
    </div>
  </section>

  <!--===========About Us===============-->

  <section class="about">
    <div class="about-img">
      <img
src="https://res.cloudinary.com/dxssqb6l8/image/upload/v1605293719/outdoor_tjjhxk
.jpg">
    </div>
    <div class="about-text">
      <small>ABOUT OUR COMPANY</small>
      <h2>We PLAN YOUR WAY to most beautiful places that you can ever ask
for!</h2>
      <p>We help the travellers to plan their best customizied itinerary in one touch
</p>


      <a href="#">ABOUT US</a>
    </div>
  </section>

  <!--===========Footer================-->
  <div class="footer">
    <div class="links">
      <h3>Quick Links</h3>
      <ul>
        <li>Recommend</li>
        <li>Explore</li>
        <li>Contact Us</li>
        <li>About</li>
      </ul>
    </div>
    <div class="links">
      <h3>Support</h3>
      <ul>
        <li>Frequently Asked Questions</li>
        <li>Report a Payment Issue</li>
        <li>Terms & Conditions</li>
        <li>Privacy Policy</li>
```

```
    </ul>
   </div>
  </div>
</body>
</html>
```

- **Result.html**

```
<!--================Travel Itinerary================-->

<section class="itinerary">
  <div class="itinerary-heading">
    <h1>Travel Itinerary</h1>
  </div>
  <div class="itinerary-details">

   {% for day, destinations in result.items %}
 <section class="itinerary">
  <h2>{{ day }}: Activity</h2>
  <ul>
    {% for destination in destinations %}
      <li>{{ destination }}</li>
    {% endfor %}
  </ul>
 </section>
{% endfor %}

    <!-- Repeat for each day and destination -->
  </div>
 </section>
```

- **Recommend.html**

```
<!--===========banner1=================-->

 <section class="image">
  <div>
    <h1> Explore the world you live !!</h1>
  </div>
  <div class="contact-wrap">
    <div class="contact-in">
       <h1>Contact Info</h1>
       <h2><i class="fa fa-phone" aria-hidden="true"></i> Phone</h2>
       <p>123-456-789</p>
       <h2><i class="fa fa-envelope" aria-hidden="true"></i> Email</h2>
       <p>info@democompany.com</p>
       <h2><i class="fa fa-map-marker" aria-hidden="true"></i> Address</h2>
```

```
            <p>Vasant Vihar, Delhi, India</p>
            <ul>
               <li><a href="#"><i class="fa fa-facebook" aria-hidden="true"></i></a></li>
               <li><a href="#"><i class="fa fa-twitter" aria-hidden="true"></i></a></li>
               <li><a href="#"><i class="fa fa-google" aria-hidden="true"></i></a></li>
               <li><a href="#"><i class="fa fa-instagram" aria-
hidden="true"></i></a></li>
            </ul>
         </div>
         <div class="contact-in">
            <h1>Recommend your Favourite Spot</h1>
            <form action="home">
               <input type="text" placeholder="Full Name" class="contact-in-input">
               <input type="text" placeholder="Email" class="contact-in-input">
               <input type="text" placeholder="Destination" class="contact-in-input">
               <input type="text" placeholder="City" class="contact-in-input">
               <input type="text" placeholder="State" class="contact-in-input">
               <input type="text" placeholder="Rating" class="contact-in-input">
               <textarea placeholder="Review" class="contact-in-textarea"></textarea>
               <input type="submit" value="SUBMIT" class="contact-in-btn">
            </form>
         </div>
         <div class="contact-in">
            <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d224345.839231
92776!2d77.06889754725782!3d28.52758200617607!2m3!1f0!2f0!3f0!3m2!1i1024!
2i768!4f13.1!3m3!1m2!1s0x390cfd5b347eb62d%3A0x52c2b7494e204dce!2sNew%
20Delhi%2C%20Delhi!5e0!3m2!1sen!2sin!4v1601968196548!5m2!1sen!2sin"
width="100%" height="auto" frameborder="0" style="border:0;" allowfullscreen=""
aria-hidden="false" tabindex="0"></iframe>
         </div>
      </div>
```

- **Explore.html**

```
<!--===========banner1=================-->

   <section class="banner">
      <div class="banner-text-item">
   <div class ="row">
      <div class="col">
         <h2>Explore the essence of South India</h2>
      </div>
      <div class="cards">
         <div class="states">
         <div class="card1">
            <h5>Kerala</h5>
        </div>
         <div class="card2">
```

```
                <h5>Tamil Nadu</h5>
            </div>
            <div class="card3">
                <h5>Karnataka</h5>
            </div>
            <div class="card4">
                <h5>Andhra Pradesh</h5>
            </div>
            </div>
        </div>
    </div>

</section>
```

5. TESTING

5.1. TEST CASES

The following test scenarios have been identified:

(1) Signup credentials

(a) Entering a new username and email, which is not present in the database.

(b) Entering a username and email already existing in the database

(c) Incorrect email format

(2) Login credentials

(a) Login using correct username and password

(b) Login using incorrect username and password

(3) Recommendation System

(a) Entering all the necessary details into generate the travel itinerary

(b) Submitting without filling up mandatory details

Table 5.1 Test Cases

SL No.	Module Name	Test Case No.	Test Case Description	Expected Result
1	Registration	TC1	The user register using valid credentials	The user signup should be successful
2	Registration	TC2	The user signup without willing up all the fields	An alert message to fill up the field should be displayed
3	Registration	TC3	The user signup using the existing username	An error message should be displayed while signing up
4	Registration	TC4	The user signup using existing email	An error message should be displayed while signing up
5	Login	TC5	Login using valid credentials	The user should be logged in successfully
6	Login	TC6	Login using incorrect password	An error message should be displayed while login
7	Login	TC7	Login using incorrect username	An error message should be displayed while login

8	Recommend	TC8	Fill all the fields to generate the itinerary	The user should be provided with the travel itinerary generated based on the user input
9	Recommend	TC9	Fill all the fields except the visit date	The user should be provided with the travel itinerary generated based on the user input
10	Recommend	TC10	Fill all the fields except anyone of the mandatory field	An alert message to fill up the field should be displayed

5.2. TESTING APPROACHES

The scope of testing is to test the operating characteristics of the application. The primary focus of this test plan is on the usability and functionality of the app. The following testing strategies have been applied on the web application:

1. Unit Testing: The main objective of unit testing is to verify whether every single unit works as intended. The application was divided into 5 main modules which included registration and login, user input page, recommendation page and explore. All the modules that were implemented were tested separately and all bugs which were found were reviewed and later rectified.

2. System Testing: This test was conducted after the unit testing. The objective of System Testing is to evaluate the compliance of an integrated application with its requirements.
We Conducted the following System Tests:

● **Interface Testing:** The application was tested in multiple web browsers in multiple systems. The UI is being loaded correctly according to the design and it is being able to connect to the database and retrieve correct information without any failures.
● **Usability Testing:** Usability testing is to check whether the application is easy to use and understand from the user's point of view. All the font size and buttons were rechecked to see if it was clear and visible to the average user. Every input field and button have a label attached to it for better understanding of the application by the user.
Fields Testing: All the required fields in different modules of the application were tested and verified that it works correctly. The mandatory fields show an error if the user tries to proceed without filling in the information required.

● **Database Testing:** Whenever users use this application simultaneously it can connect correctly to the cloud database and perform CRUD operations. This information has been tested and results in an update of information at real time.

● **Compatibility Testing:** This application was tested in multiple browsers to ensure that it displays and works correctly according to the design specified. Since we are using simple HTML and CSS for its design. It is compatible with all standard web browsers.

3. Security Testing: The data of users of the application including their passwords are protected from any type of attacks. This was ensured by using the inbuilt AuthenticateForm in Django where the password is hashed before storing in the database. All reading and writing to the database can only happen when the user has verified their login through authentication.

5.3. TEST REPORTS

Table 5.2 Test Reports

SL No	Test Case Number	Test Status	Test Report
1	TC1	Successful	Fig 5.3
2	TC2	Successful	Fig 5.4
3	TC3	Successful	Fig 5.5
4	TC4	Successful	Fig 5.6
5	TC5	Successful	Fig 5.7
6	TC6	Successful	Fig 5.8
7	TC7	Successful	Fig 5.8
8	TC8	Successful	Fig 5.9
9	TC9	Successful	Fig 5.9
10	TC10	Successful	Fig 5.10

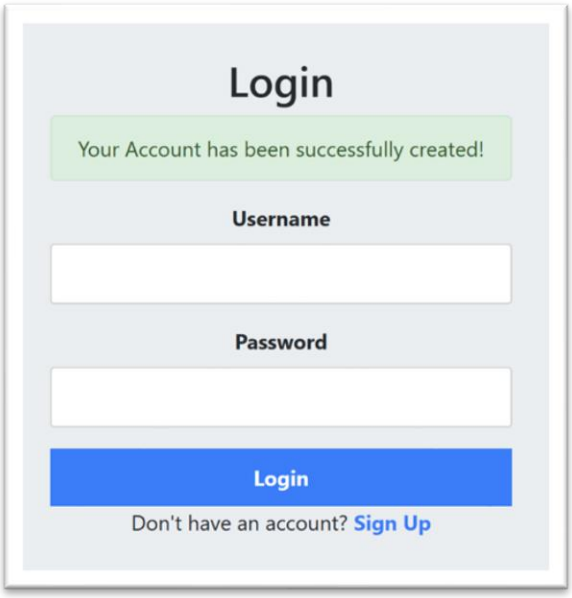


Fig 5.3 TC1

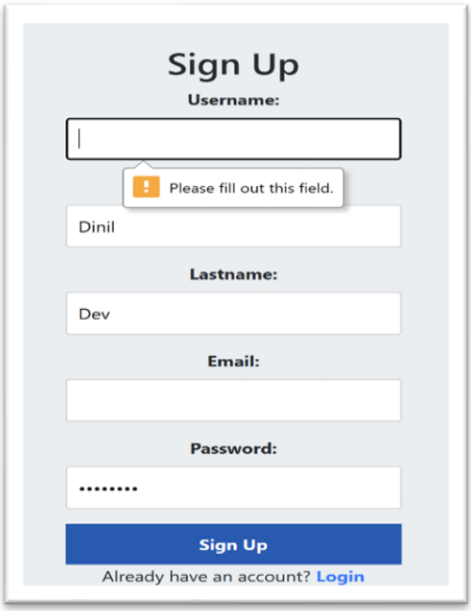


Fig 5.4 TC2

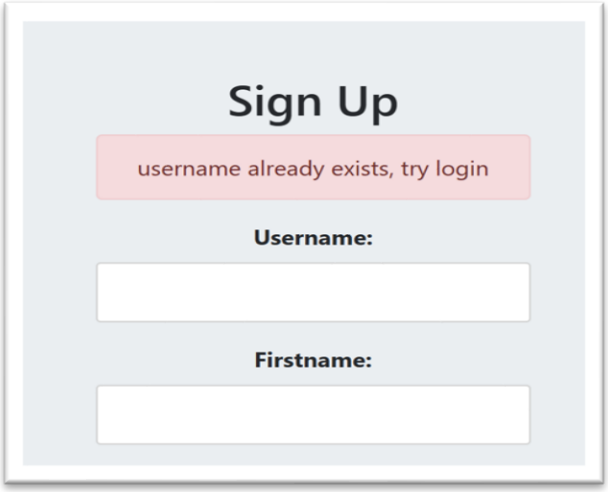


Fig 5.5 TC3

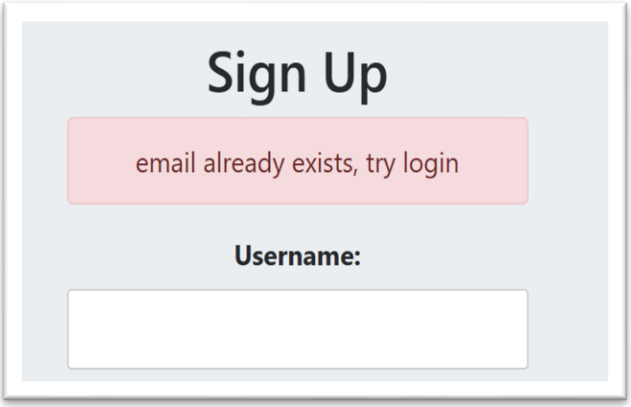


Fig 5.6 TC4

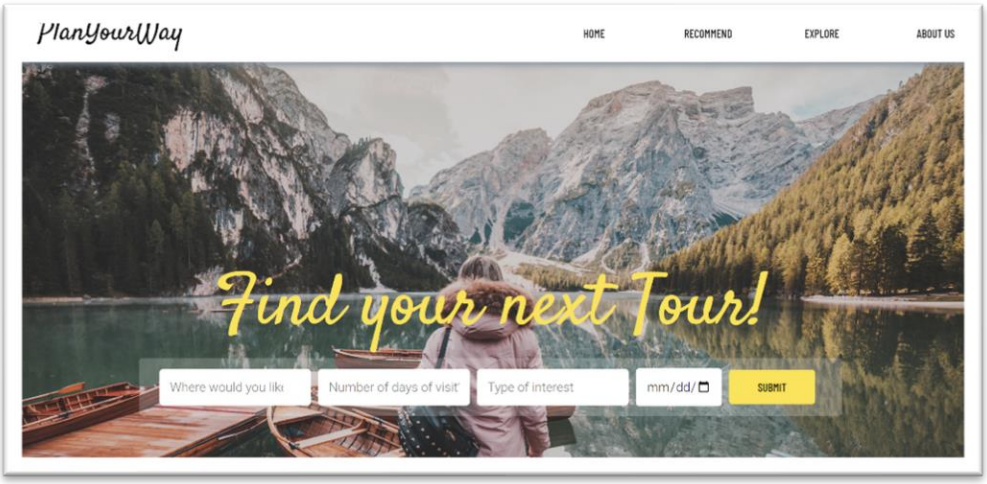


Fig 5.7 TC5

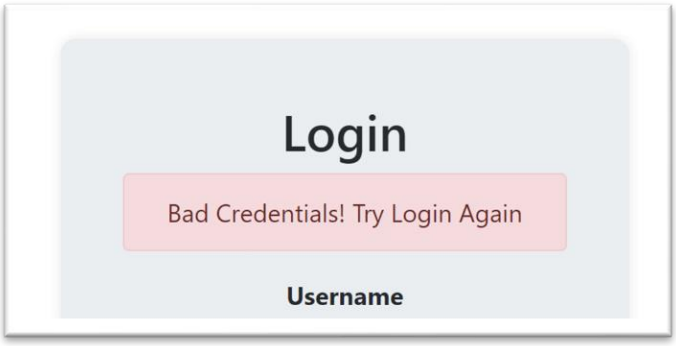


Fig 5.8:TC6 & TC7

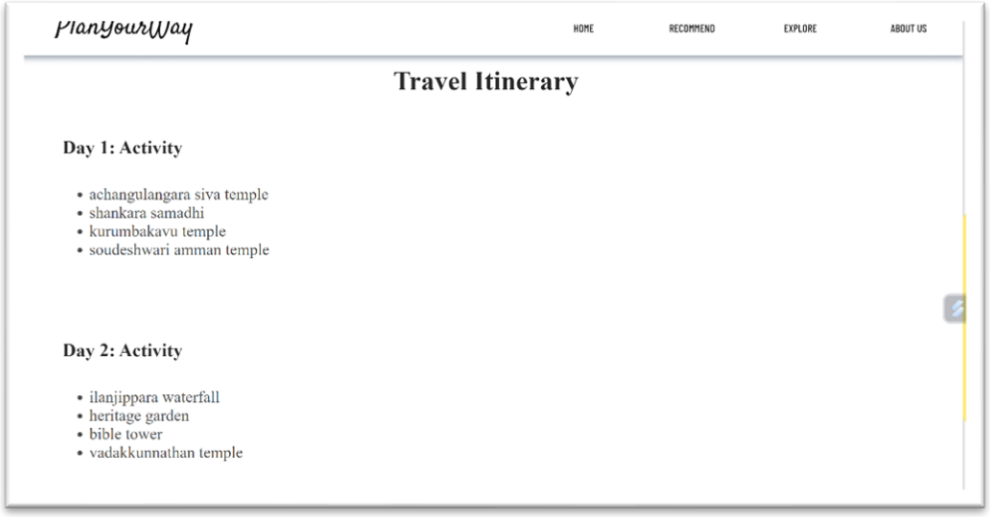


Fig 5.9 TC8 & TC9

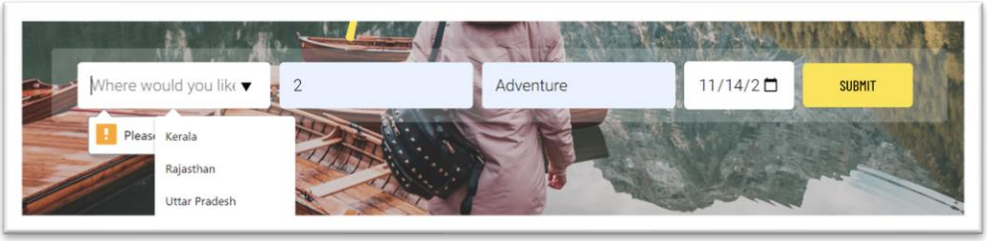


Fig 5.10 TC10

6. CONCLUSION

This is the final chapter which explains all the design and implementation issues faced during the design and implementation of the project. The chapter explains how the problems identified with the different existing systems have been resolved, the limitations and the future enhancements of the project.

6.1 DESIGN AND IMPLEMENTATION ISSUES

1. The time taken to load the distance matrix is high, since it requires API making data request and waiting for Bing Maps to respond. This will result in the process of Itinerary generation slow.
2. The application requires fast and stable internet connection to run the application properly.
3. The condition-based recommendation may not be the optimal solution. Also, Python is used for implementing, which is not the fastest language around, leading to slow websites. As the application is built with the Django framework it uses Python in its backend.
4. Since the Model View Template architecture of the Django framework is followed in this project, when an error is displayed in the webpage it is sometimes difficult to trace to the cause of the error.

6.2 ADVANTAGES AND DISADVANTAGES

6.2.1 Advantages

- The use of recommendation helps in taking the most popular tourist attractions and distance optimization helps in saving time and money thereby giving a smooth and efficient journey for the travellers.
- The application collects user feedback and reviews, enabling continuous improvement in the recommendations and services, ensuring that users receive the most up-to-date and valuable information.
- The process of travel planning is time consuming and tedious. The application helps in automating this process and gives users an effortless trip planning.

6.1.2 Limitations

- The itinerary generation is slow due to distance matrix calculation using Bing Maps API.

- Time constraint is not considered due to unavailability of data, it is assumed that the user, on an average will visit 4 destinations in a day.
- Interstate travel is not in the scope of this project.

6.3. FUTURE SCOPE OF THE PROJECT

The application has a very vast scope in the future. Updates to the application can be made in order to increase its efficiency and accuracy. The following are the future enhancements that can be done:

- Accommodation and time spent on an average detail can be added, which will then provide the users with complete travel itinerary. The website can also make suggestions to the user for choice of accommodation, providing similar hotels and the link to their websites. The users can then book the hotel of their choice.
- The itinerary once generated, cannot be customized. This can be helpful to the users if they want to make slight changes in the itinerary. Allowing the users to customize the generated itinerary and then optimize the changed destinations again, providing a new ordered list of destinations can be done in the future. Further, the website can also let the users download the itinerary generated as per their needs.
- The web application can be extended to suggest itineraries for users by detecting their live location. If the user inputs their current location, then the system should be able to detect places nearby and make itineraries with different types of interest provided by the user.
- Interstate travel can be included if there is adequate data available. Also, the scope of this project currently is limited to road transport. Provided that data is available, this can be extended to air and water transport, which is also common interstate transports.

7. REFERENCES

[1] Ivan Hintoro, Sandy Kosasi, David, Susanti Margaretha Kuway and Gat., "Genetic Algorithm Application in Route Optimization Using Google Maps API," *Research gate Publication* . DOI: 10.33050/ccit.v16i2

[2] Jin, Huang, "Tourism Route Planning of Cultural Heritage Scenic Spots Based on Genetic Simulation Ant Colony Algorithm", *Frontier Computing (pp.1482-1487)* May 2023. DOI: 10.1007/978-981-99-1428-9_194

[3] Jingwen Qi and Qiuhong Wang, "Tourism Route Selection Model for Tourism Sustainable Development Based on Improved Genetic Algorithm," *International Transactions on Electrical Energy Systems Volume 2022, Article ID 4287011, 10 pages* doi.org/10.1155/2022/4287011

[4] Na Lou, "Analysis of the Intelligent Tourism Route Planning Scheme Based on the Cluster Analysis Algorithm," *Computational Intelligence and Neuroscience Volume 2022, Article ID 3310676, 10 pages* doi.org/10.1155/2022/3310676

[5] B.-M. Hsu, "Design of Travel Itinerary Planning System Based on Artificial Intelligence," *ICAIIT 2020 Journal of Physics: Conference Series* 1533 (2020) 032078 doi:10.1088/1742-6596/1533/3/032078

[6] S.Bairavel, Dr. M. Krishnamurthy, "User Preference and Reviews Analysis with Neural Networks for Travel Recommender Systems," *International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 8 (2020), pp. 1896-1900* https://dx.doi.org/10.37624/IJERT/13.8.2020.1896-1900